

Modellistica Medica

Maria Grazia Pia

INFN Genova

Scuola di Specializzazione in Fisica Sanitaria

Genova

Anno Accademico 2002-2003

Lezione 24-25

Modeling interactions of particles with matter

Overview of Geant4

The role of simulation

- Simulation plays a fundamental role in various domains and phases of an experimental physics project
 - design of the experimental set-up
 - evaluation and definition of the potential physics output of the project
 - evaluation of potential risks to the project
 - assessment of the performance of the experiment
 - development, test and optimisation of reconstruction and physics analysis software
 - contribution to the calculation and validation of physics results
- The scope of these lectures (and of Geant4) encompasses the simulation of the passage of particles through matter
 - there are other kinds of simulation components, such as *physics event generators*, *electronics response* generation, etc.
 - often the simulation of a complex experiment consists of several of these components interfaced to one another

Domains of application

- **HEP, nuclear, astrophysics and astro-particle physics experiments**
 - the most “traditional” field of application
- **Radiation background studies**
 - evaluation of safety constraints and shielding for the experimental apparatus and human beings
- **Medical applications**
 - radiotherapy
 - radiodiagnostics
- **Biological applications**
 - radiation damage (in human beings, food etc.)
- **Space applications**
 - Radiation and biological damage to equipment and astronauts

Basic requirements

- Modeling the experimental set-up
- Tracking particles through matter
- Interaction of particles with matter
- Modeling the detector response
- Run and event control

- Accessory utilities (*random number generators, PDG particle information etc.*)
- Interface to event generators
- Visualisation of the set-up, tracks and hits
- User interface
- Persistency

Fast and full simulation

- Usually in a typical HEP experiment there are two types of simulations
- **Fast simulation**
 - mainly used for feasibility studies and quick evaluations
 - coarse set-up description and physics modeling
 - usually directly interfaced to event generators
- **Full simulation**
 - used for precise physics and detector studies
 - requires a detailed description of the experimental set-up and a complex physics modeling
 - usually interfaced to event generators and event reconstruction
- *Traditionally fast and full simulation are done by different programs and are not integrated in the same environment*
 - complexity of maintenance and evolution
 - possibility of controversial results

The zoo

EGS4, EGS5, EGSnrc

MCNP, MCNPX, A3MCNP, MCNP-DSP, MCNP4B

Penelope

Geant3, Geant4

Tripoli-3, Tripoli-3 A, Tripoli-4

Peregrine

MVP, MVP-BURN

MARS

...and I probably forgot some more

Many codes not publicly distributed

A lot of business around MC

NMTC

HERMES

FLUKA

EA-MC

DPM

SCALE

GEM

MF3D

MCU

MORSE

TRAX

MONK

MCBEND

VMC++

LAHET

RTS&T-2000

Integrated suites vs. specialised codes

Specialised packages cover a specific simulation domain

Pro:

- the specific issue is treated in great detail
- sometimes the package is based on a wealth of specific experimental data
- simple code, usually relatively easy to install and use

Contra:

- a typical experiment covers many domains, not just one
- domains are often inter-connected

Integrated packages cover all/many simulation domains

Pro:

- the same environment provides all the functionality

Contra:

- it is more difficult to ensure detailed coverage of all the components at the same high quality level
- monolithic: take all or nothing
- limited or no options for alternative models
- usually complex to install and use
- difficult maintenance and evolution

The Toolkit approach

A toolkit is a set of compatible components

- each component is specialised for a specific functionality
- each component can be refined independently to a great detail
- components can be integrated at any degree of complexity
- components can work together to handle inter-connected domains
- it is easy to provide (and use) alternative components
- the simulation application can be customised by the user according to his/her needs
- maintenance and evolution - both of the components and of the user application - is greatly facilitated

...but what is the price to pay?

- the user is invested of a greater responsibility
- he/she must critically evaluate and decide what he/she needs and wants to use

Geant4 development

Requirements
R&D phase
Production phase

Geant as simulation infrastructure

- Geant is a simulation tool, that provides a general infrastructure for
 - the description of geometry and materials
 - particle transport and interaction with matter
 - the description of detector response
 - visualisation of geometries, tracks and hits
- The user develops specific code for
 - the primary event generator
 - the geometrical description of the set-up
 - the digitisation of the detector response

The historical background

• Geant3

- has been used by most HEP experiments
- used also in nuclear physics experiments, medical physics, radiation background studies, space applications etc.
- frozen since March 1994 (Geant3.21)
- ~200K lines of code
- equivalent of ~50 man-years, along 15 years

• The result is a complex system

- because its problem domain is complex
- because it requires flexibility for a variety of applications
- because its management and maintenance are complex

• It was not self-sufficient

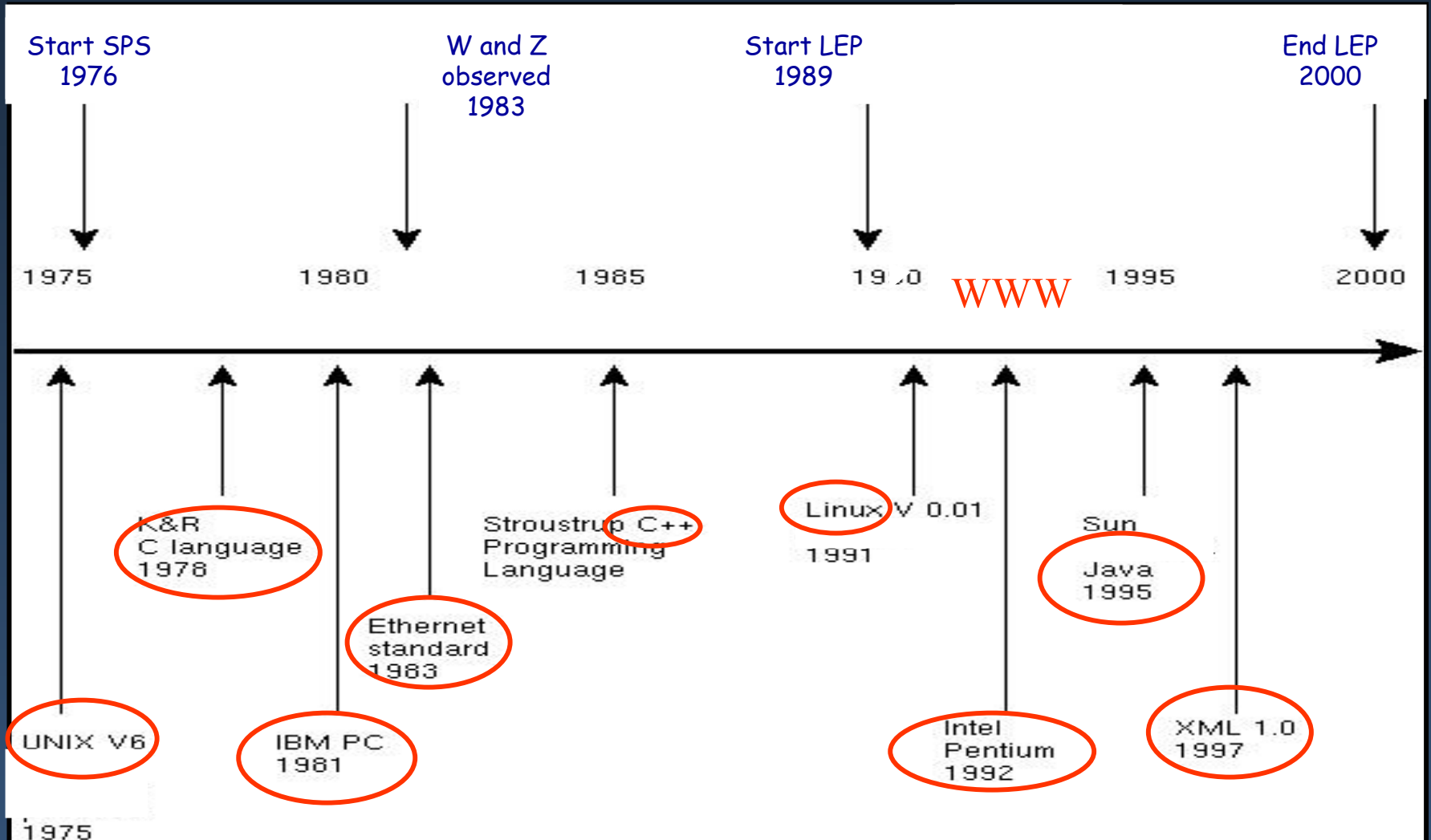
- hadronic physics is not native, it is handled through the interface to external packages

New simulation requirements

- Very high statistics to be simulated
 - robustness and reliability for large scale production
- Exchange of CAD detector descriptions
- Transparent physics for experimental validation
- Physics extensions to high energies
 - LHC, cosmic ray experiments...
- Physics extensions to low energies
 - space science, astrophysics, medical physics, nuclear and atomic physics...
- Reliable hadronic physics
 - not only for calorimetry, but also for PID applications (CP violation experiments)
- ...etc.
- *User requirements formally collected and coded according to PSS05*

Geant4 URD

The environment: the last 25 years



OO technology



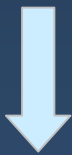
Openness to extension and evolution

- thanks to **polymorfism** and **dynamic binding**, new implementations (models, algorithms etc.) can be added without changing the existing code

Robustness and ease of maintenance

- thanks to **encapsulation**, objects have crisp boundaries
- minimisation of coupling, through well defined protocols and dependencies

Strategic vision



Toolkit approach

A toolkit is a set of compatible **components**

- each component is **specialised** for a specific functionality
- each component can be **refined** independently to a great detail
- components can be **integrated** at any degree of complexity
- it is easy to provide (and use) **alternative** components
- the simulation application can be **customised** by the user according to his/her needs
- **maintenance** and **evolution** - both of the components and of the user application - is greatly facilitated



RD44

Approved as R&D at CERN, end 1994

- > 100 physicists and software engineers
- ~ 40 institutes, international collaboration
- responded to DRCC/LCB

● Milestone: end 1995

- OO methodology, problem domain analysis, full OOAD
- tracking prototype, performance evaluation

● Milestone: spring 1997

- α -release with same functionality as Geant 3.21
- persistency (hits), ODBMS
- transparency of physics models

● Milestone: July 1998

- public β -release

● Milestone: end 1998

- production release: Geant4 0.0
- **end of RD44**

Since January 1999:

- International Geant4 Collaboration
- 2 Geant4 releases/year



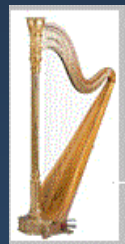
Geant4 Collaboration



MoU based

Distribution, Development and User Support of Geant4

- Atlas, BaBar, CMS, HARP, LHCb
- CERN, JNL, KEK, SLAC, TRIUMF
- ESA, INFN, IN2P3, PPARC
- Frankfurt, Barcelona, Karolinska, Lebedev
- COMMON (Serpukov, Novosibirsk, Pittsburg, Northeastern, Helsinki, TERA etc.)
- Collaboration Board
 - manages resources and responsibilities
- Technical Steering Board
 - manages scientific and technical matters
- Working Groups
 - do maintenance, development, QA, etc.



Members of National Institutes, Laboratories and Experiments participating acquire the right to the Production Service and User Support

For others: free code and user support on best effort basis



Budker Inst. of Physics
IHEP Protvino
MEPHI Moscow
Pittsburg University



Maria Grazia Pia, INFN Genova

Modellistica Medica, Scuola di Specializzazione in Fisica Sanitaria, Univ. Genova

What is **Geant 4**?

- **OO Toolkit for the simulation of next generation HEP detectors**
 - ...of the current generation too
 - ...not only of HEP detectors
 - already used also in nuclear physics, medical physics, astrophysics, space applications, radiation background studies etc.
- It is also an experiment of distributed software production and management, as a large international collaboration with the participation of various experiments, labs and institutes
- It is also an experiment of application of rigorous software engineering methodologies and Object Oriented technologies to the HEP environment

The Geant 4 kit

• Code

- ~1M lines of code
- continuously growing
- publicly downloadable from the web

• Documentation

- 6 manuals
- publicly available from the web

• Examples

- distributed with the code
- navigation between documentation and examples code
- various complete applications of (simplified) real-life experimental set-ups

• Platforms

- Linux, SUN (*DEC, HP*)
- Windows-NT: Visual C++

• Commercial software

- none required
- can be interfaced

• Free software

- CVS
- gmake, g++
- CLHEP

• Graphics & (G)UI

- OpenGL, OpenInventor, DAWN, VRML...
- OPACS, GAG, MOMO...

• Persistency

- it is possible to run in transient mode
- in persistent mode use a HepDB interface, ODMG standard

User support

- The Geant4 User Support covers the
 - provision of help and analysis of code-related problems
 - the consultancy
 - the requests for enhancement or new developments
 - the investigation of anomalous results
- The User Support is provided by the Geant4 Collaboration
 - See Geant4 web for details

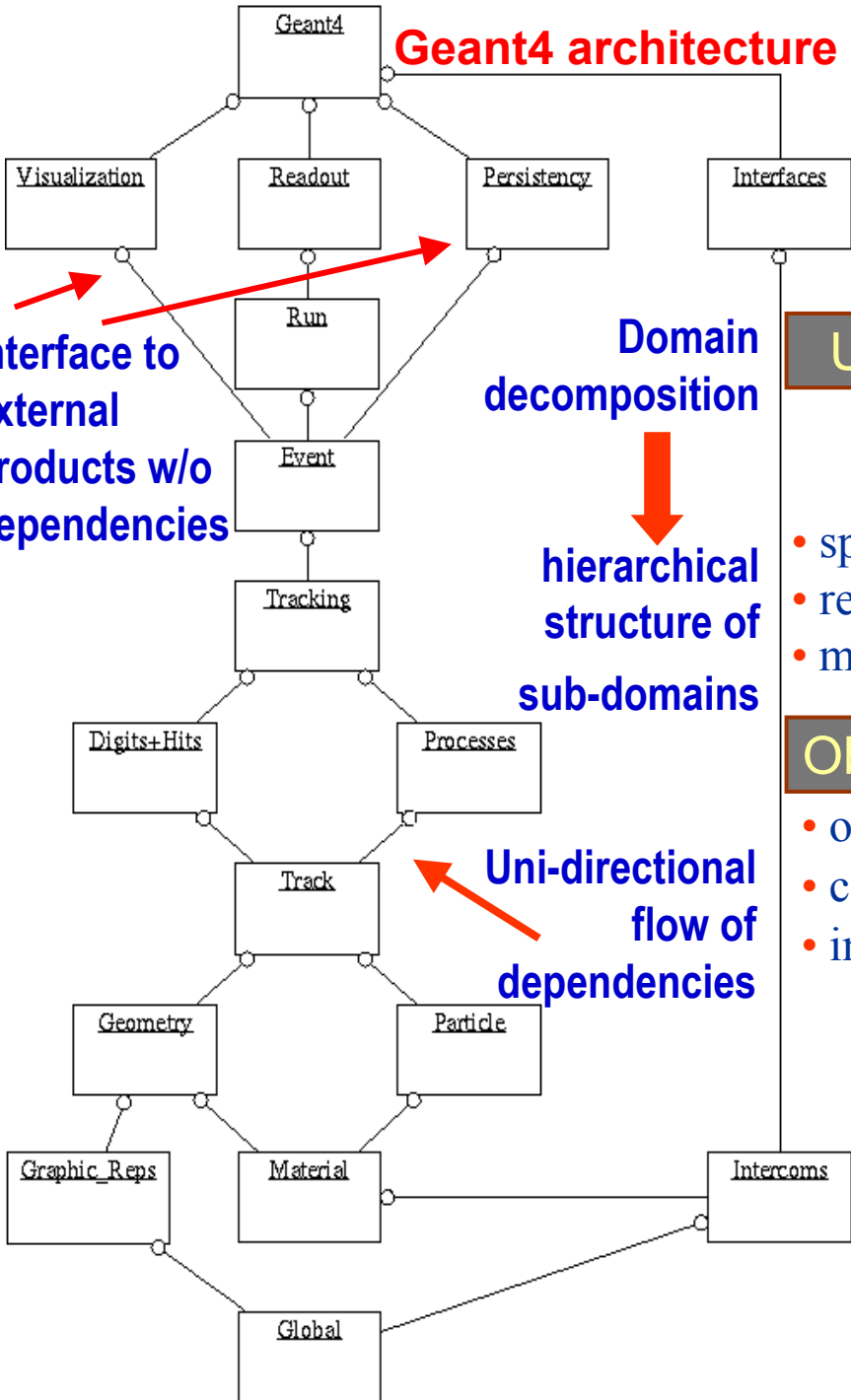
The foundation

What characterizes Geant4
Or: the fundamental concepts, which all the
rest is built upon

Physics

From the Minutes of LCB (LHCC Computing Board) meeting on 21 October, 1997:

"It was noted that experiments have requirements for independent, alternative physics models. In Geant4 these models, differently from the concept of packages, allow the user to understand how the results are produced, and hence improve the physics validation. Geant4 is developed with a modular architecture and is the ideal framework where existing components are integrated and new models continue to be developed."



Geant4 architecture

Software Engineering

plays a fundamental role in Geant4

User Requirements

- formally collected
- systematically updated
- PSS-05 standard

Software Process

- spiral iterative approach
- regular assessments and improvements (SPI process)
- monitored following the ISO 15504 model

Object Oriented methods

- OOAD
- use of CASE tools

- openness to extension and evolution
- contribute to the transparency of physics
- interface to external software without dependencies

Quality Assurance

- commercial tools
- code inspections
- automatic checks of coding guidelines
- testing procedures at unit and integration level
- dedicated testing team

Use of Standards

- de jure and de facto

Basic concepts

(to be detailed in the following lectures)

Overview of the functionalities
Kernel
System of units

System of Units

- Geant4 is independent from any system of units
- All numerical quantities are expressed with their units explicitly
- The user is not constrained to using any specific system of units in his/her application

Have you heard of the “incident” with NASA’s Mars Climate Orbiter (\$125 million)?

The kernel

• Run and event

- multiple events
 - possibility to handle the pile-up
- multiple runs in the same job
 - with different geometries, materials etc.
- powerful stacking mechanism
 - three levels by default: handle trigger studies, loopers etc

• Tracking

- decoupled from physics: all processes handled through the same abstract interface
- tracking is independent from particle type
- it is possible to add new physics processes without affecting the tracking

Description of the experimental set-up

- **Materials**

- Different kinds of materials can be defined

- **Modeling the detectors**

- Geometry
- Electric and magnetic field

- **Detector response**

- Read-out geometry
- Hits
- Digits

Physics

- **Generating primary particles**
 - Interface to Event Generators
 - Utilities provided within the Geant4 Toolkit
- **Physics interactions**
 - Processes
 - Electromagnetic physics
standard, low energy, muons, optical
 - Hadronic physics
parameterised and theoretical models
- **Particles**
 - all PDG data
 - and more, for specific Geant4 use, like ions

User classes

Initialization classes

Invoked at the initialization

- *G4VUserDetectorConstruction*
- *G4VUserPhysicsList*

Action classes

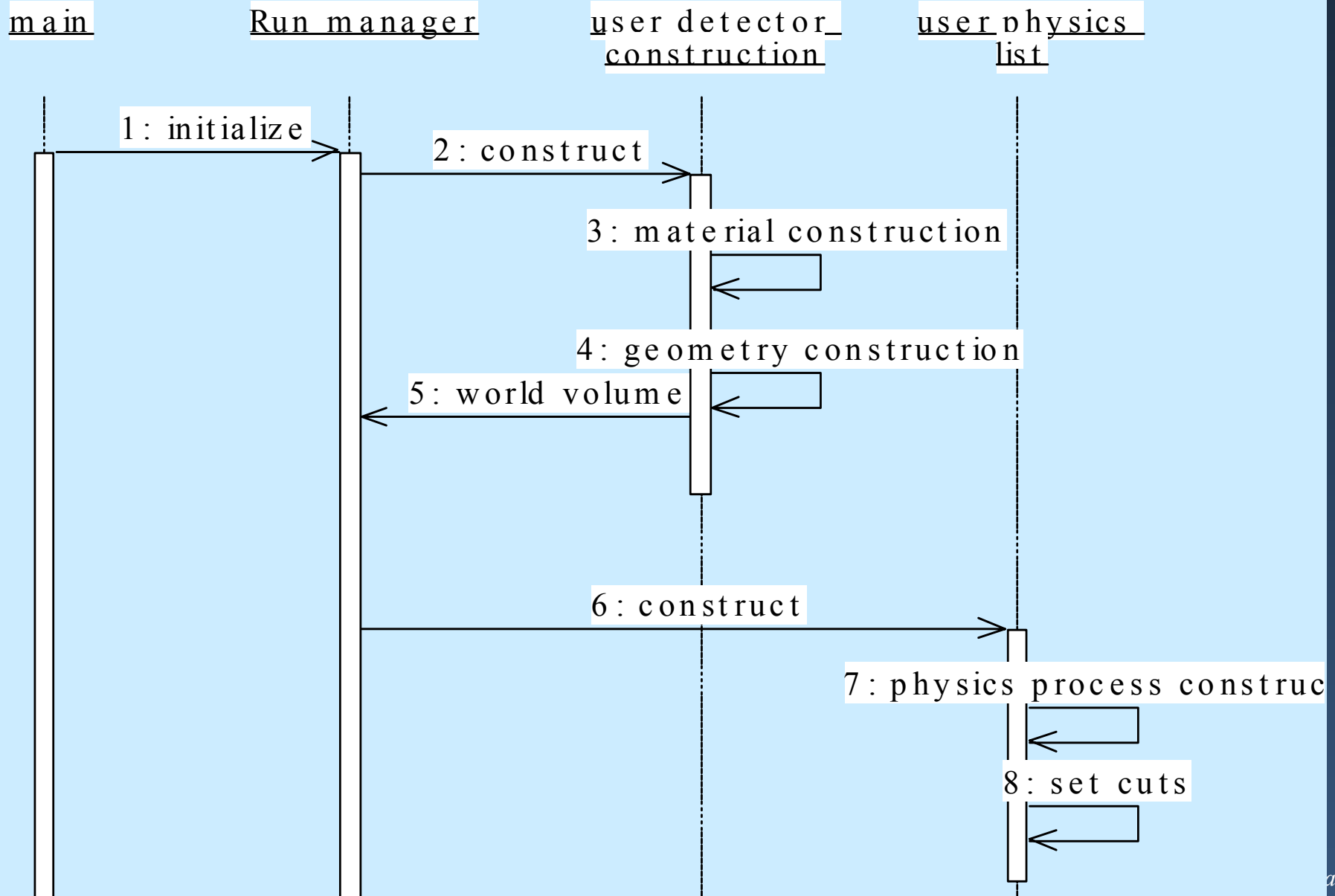
Invoked during the execution loop

- *G4VUserPrimaryGeneratorAction*
- *G4UserRunAction*
- *G4UserEventAction*
- *G4UserStackingAction*
- *G4UserTrackingAction*
- *G4UserSteppingAction*

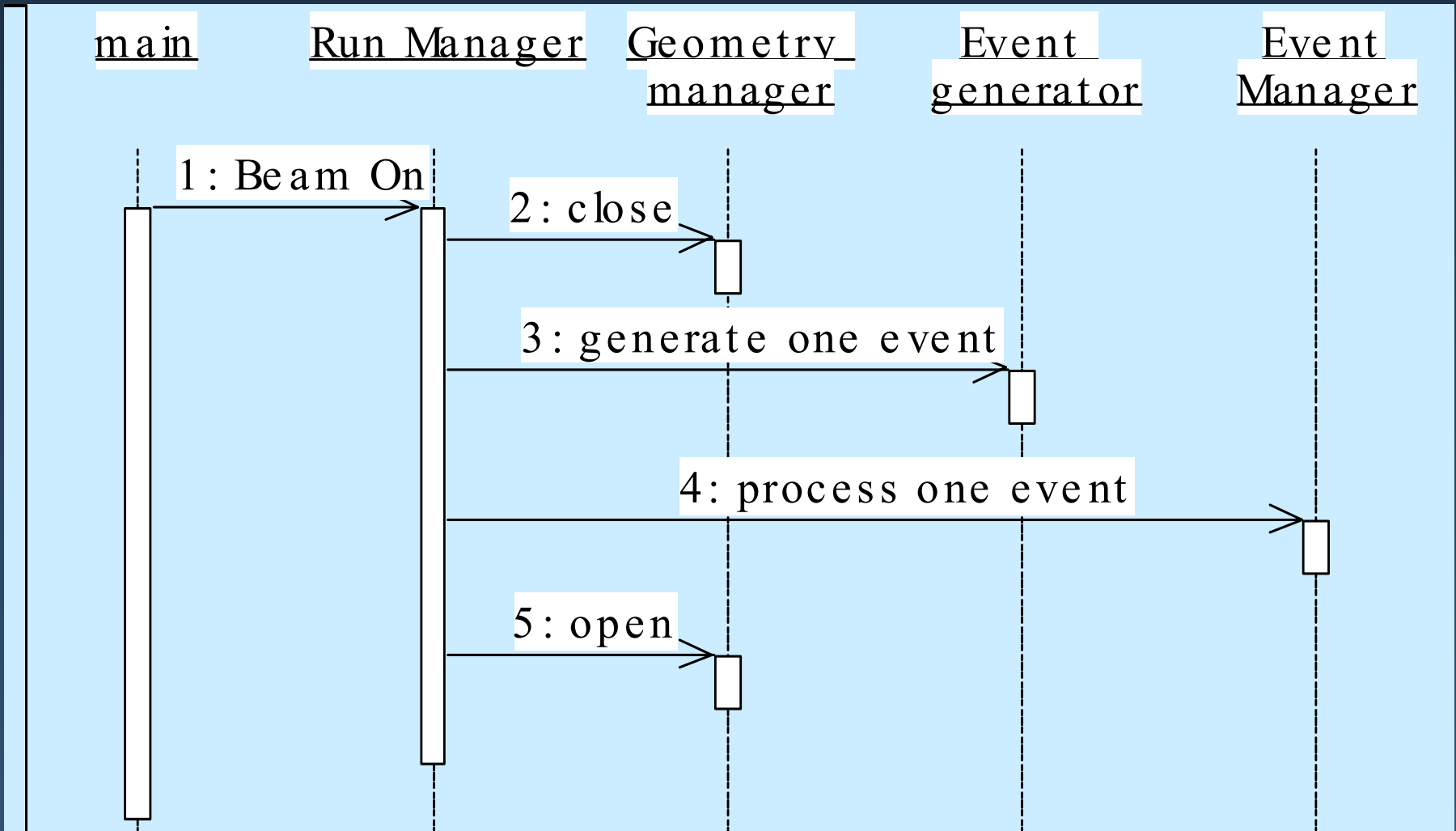
Mandatory classes:

- *G4VUserDetectorConstruction*
describe the experimental set-up
- *G4VUserPhysicsList*
select the physics you want to activate
- *G4VUserPrimaryGeneratorAction*
generate primary events

Initialisation



Beam On



Event processing

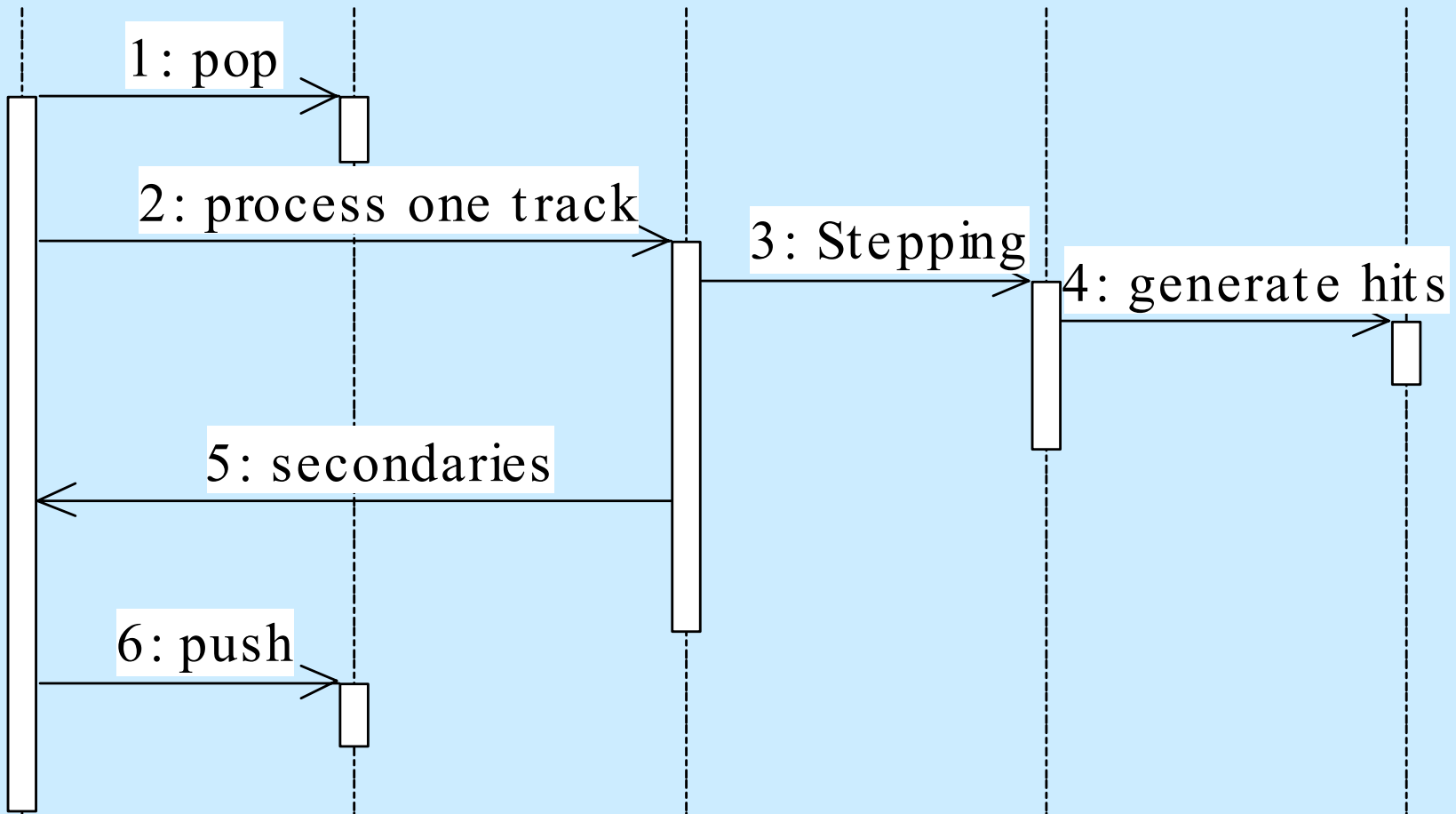
Event manager

Stacking manager

Tracking manager

Stepping manager

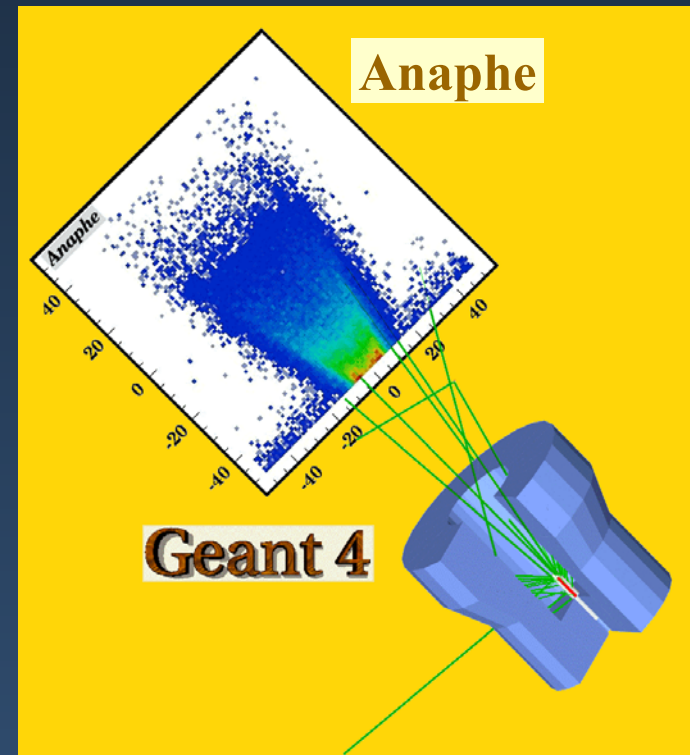
User sensitive detector



Interface to external tools in Geant4

Similar approach

- Visualisation
- (G)UI
- Persistency
- Analysis



Special modes

Will not be covered in the following lectures
Documented in User Guide for Application Developers

- Fast simulation

- Geant4 allows to perform full and fast simulation in the same environment
- The *parameterisation process* produces a direct detector response, from the knowledge of particle and volume properties

- Event biasing

- Various variance reduction techniques available

The functionality

What Geant4 can do
How well it does it

The kernel

Run and event

- multiple events
 - possibility to handle the pile-up
- multiple runs in the same job
 - with different geometries, materials etc.
- powerful stacking mechanism
 - three levels by default: handle trigger studies, loopers etc.

Geant4 has only production thresholds, **no tracking cuts**

- *all particles are tracked down to zero range*
- *energy, TOF ... cuts can be defined by the user*

Tracking

- decoupled from physics: all processes handled through the **same abstract interface**
- tracking is independent from particle type
- it is possible to add new physics processes without affecting the tracking

Materials

- Different kinds of materials can be defined

- isotopes



G4Isotope

- elements



G4Element

- molecules



G4Material

- compounds and mixtures



G4Material

- Attributes associated:

- temperature

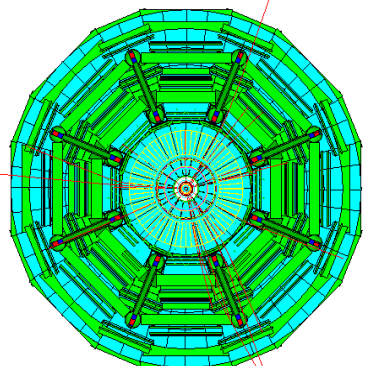
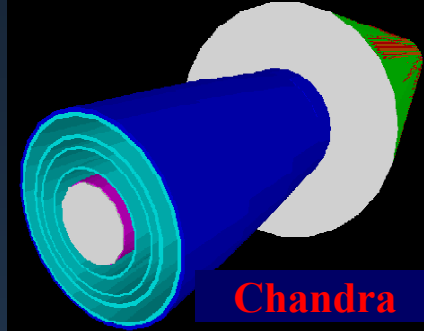
- pressure

- state

- density

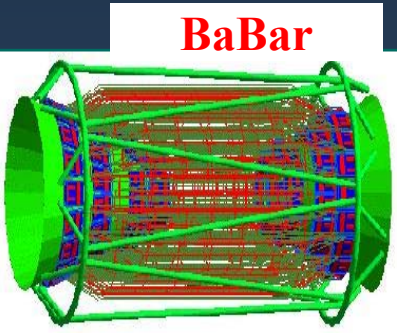
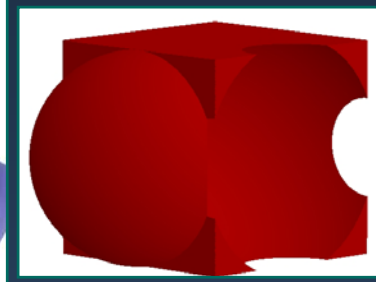
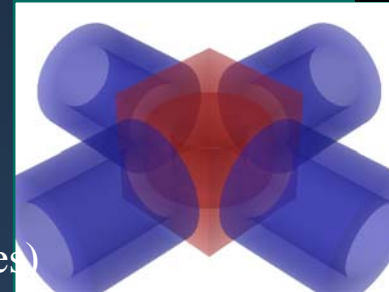
Geometry

Role: detailed detector description
and efficient navigation



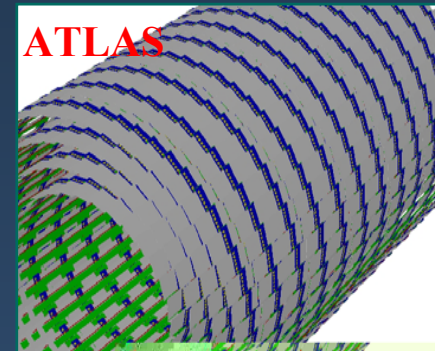
ATLAS

Multiple representations
(same abstract interface)

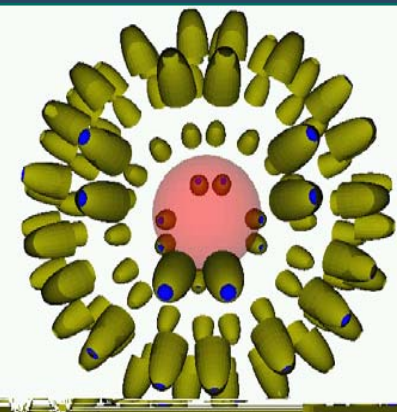


BaBar

- **CSG** (Constructed Solid Geometries)
 - simple solids
- **STEP extensions**
 - polyhedra,, spheres, cylinders, cones, toroids, etc.
- **BREPS** (**B**oundary **RE**presented **S**olids)
 - volumes defined by boundary surfaces
 - include solids defined by NURBS (*Non-Uniform Rational B-Splines*)



ATLAS

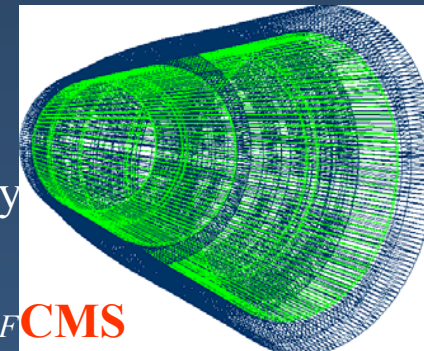


Borexino

CAD exchange: ISO STEP interface

Fields: of variable non-uniformity and differentiability

External tool for g3tog4 geometry conversion

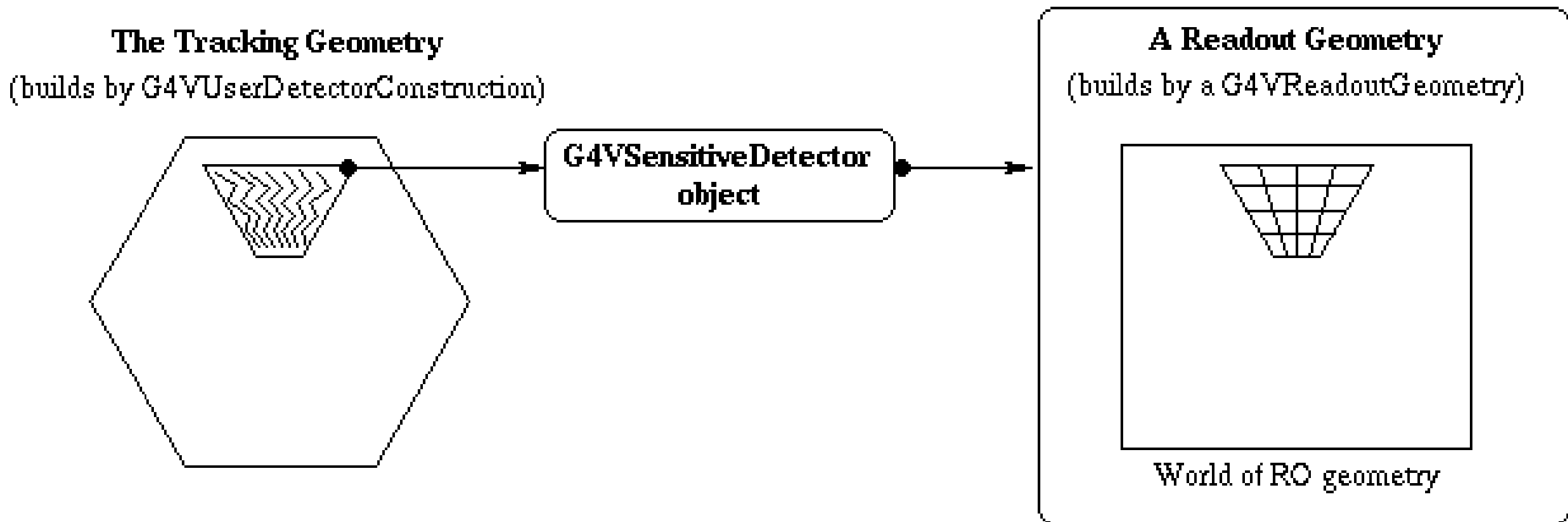


CMS

Read-out geometry

Readout geometry is a virtual and artificial geometry

- it is associated to a *sensitive detector*
- can be defined in parallel to the real detector geometry
- helps optimising the performance



Hits and Digis

- A sensitive detector creates hits using the information provided by the G4Step
- One can store various types of information in a hit
 - position and time of the step
 - momentum and energy of the track
 - energy deposition of the step
 - geometrical information
 - etc.
- A Digi represents a detector output
 - e.g. ADC/TDC count, trigger signal
- A Digi is created with one or more hits and/or other digits
- Hits collections are accessible
 - through G4Event at the end of an event
 - through G4SDManager during processing an event

Generating primary particles

• Interface to Event Generators

- through ASCII file for generators supporting /HEPEVT/
- abstract interface to Lund++

• Various utilities provided within the Geant4 Toolkit

- ParticleGun
 - beam of selectable particle type, energy etc.
- GeneralParticleSource
 - provides sophisticated facilities to model a particle source
 - used to model space radiation environments, sources of radioactivity in underground experiments etc.
- you can write your own, inheriting from *G4VUserPrimaryGeneratorAction*

• Particles

- all PDG data
- and more, for specific Geant4 use, like ions

Physics: general features

- Ample variety of physics functionalities
- Uniform treatment of electromagnetic and hadronic processes
- Abstract interface to physics processes
 - Tracking independent from physics
- Distinction between processes and models
 - often multiple models for the same physics process (complementary/alternative)
- Open system
 - Users can easily create and use their own models
- Transparency (supported by *encapsulation and polymorphism*)
 - Calculation of cross-sections independent from the way they are accessed (data files, analytical formulae etc.)
 - Distinction between the calculation of cross sections and their use
 - Calculation of the final state independent from tracking
- Modular design, at a fine granularity, to expose the physics
- Explicit use of units throughout the code
- Public distribution of the code, from one reference repository worldwide

Processes

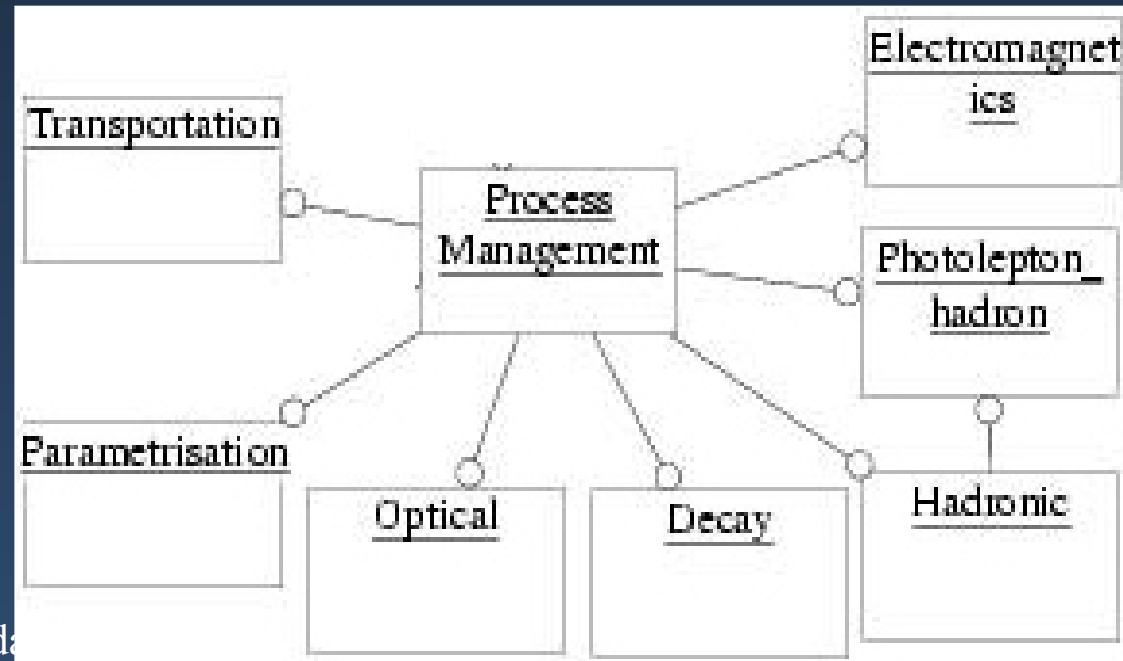
- Processes describe how particles interact with material or with a volume

- Three basic types

- **At rest** process
(eg. decay at rest)
- **Continuous** process
(eg. ionization)
- **Discrete** process
(eg. decay in flight)

- Transportation is a process

- interacting with volume bound



- A process which requires the shortest interaction length limits the step

Data libraries

- Systematic collection and evaluation of experimental data from many sources worldwide
- Databases
 - ENDF/B, JENDL, FENDL, CENDL, ENSDF, JEF, BROND, EFF, MENDL, IRDF, SAID, EPDL, EEDL, EADL, SANDIA, ICRU etc.
- Collaborating distribution centres
 - NEA, LLNL, BNL, KEK, IAEA, IHEP, TRIUMF, FNAL, Helsinki, Durham, Japan etc.
- The use of evaluated data is important for the validation of physics results of the experiments

Electromagnetic physics

energy
loss

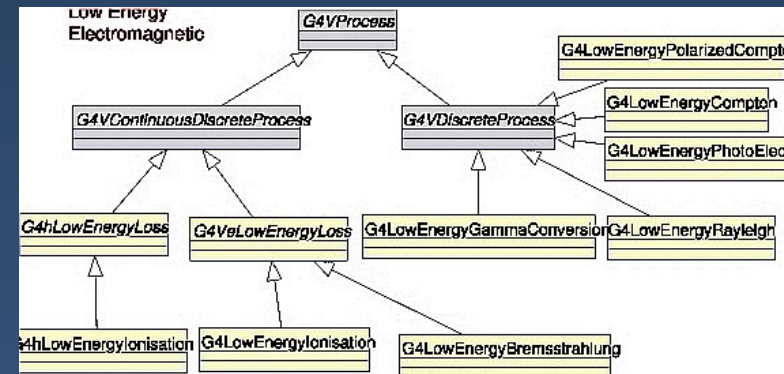
- Multiple scattering
- Bremsstrahlung
- Ionisation
- Annihilation
- Photoelectric effect
- Compton scattering
- Rayleigh effect
- γ conversion
- e^+e^- pair production
- Synchrotron radiation
- Transition radiation
- Cherenkov
- Refraction
- Reflection
- Absorption
- Scintillation
- Fluorescence
- Auger (in progress)

- electrons and positrons
- γ , X-ray and optical photons
- muons
- charged hadrons
- ions

Comparable to Geant3 already in the α release (1997)

Further extensions (*facilitated by the OO technology*)

- **High energy extensions**
 - needed for LHC experiments, cosmic ray experiments...
- **Low energy extensions**
 - fundamental for space and medical applications, dark matter and ν experiments, antimatter spectroscopy etc.
- **Alternative models for the same process**



All obeying to the same abstract Process interface \uparrow transparent to tracking

Hadronic physics

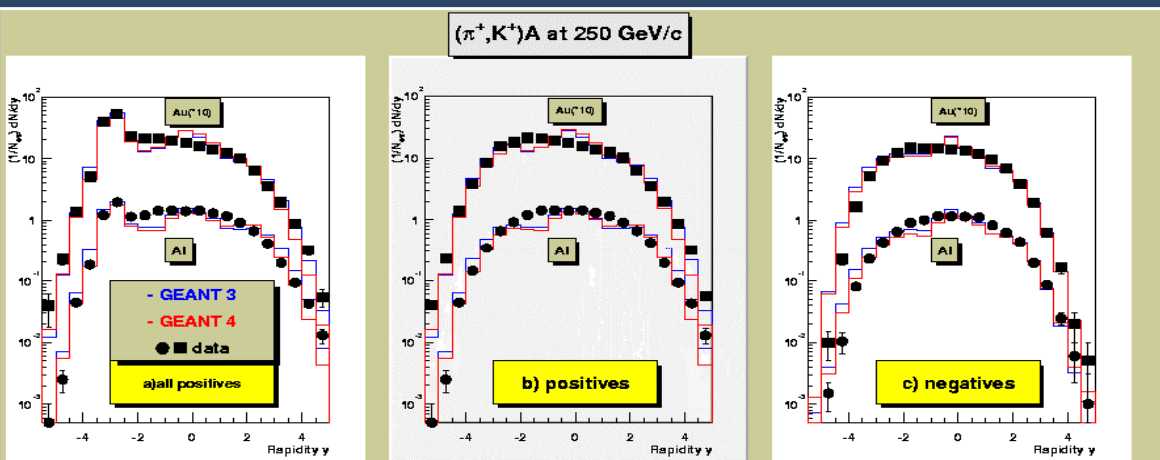
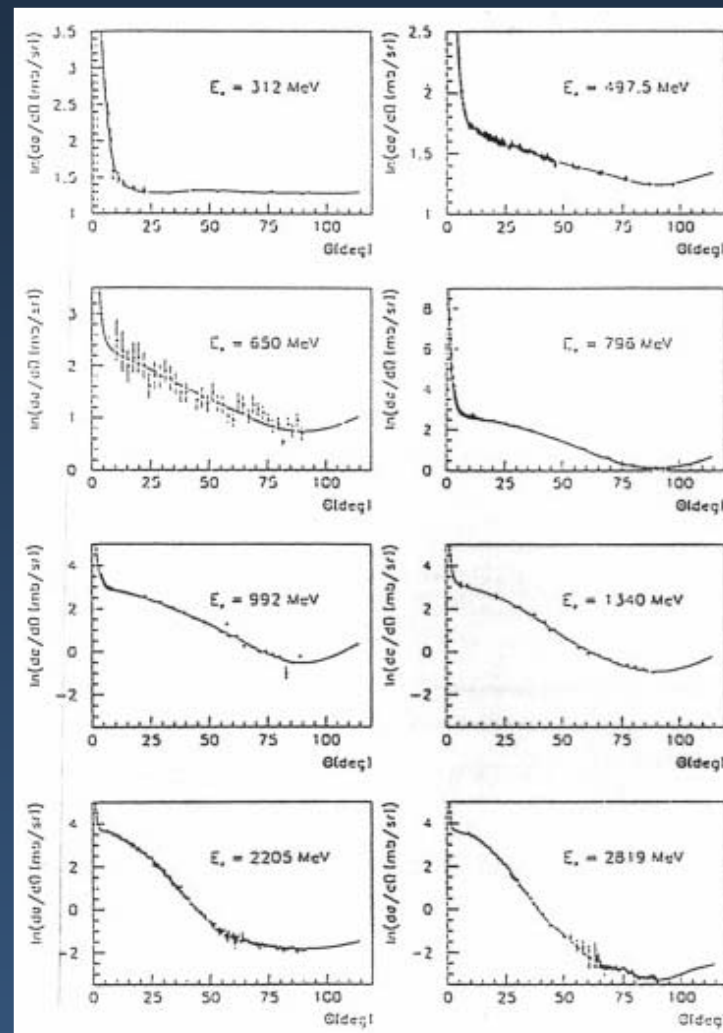
- Completely different approach w.r.t. the past (Geant3)
 - native
 - transparent
 - no longer interface to external packages
 - clear separation between data and their use in algorithms
- Cross section data sets
 - transparent and interchangeable
- Final state calculation
 - models by particle, energy, material
- Ample variety of models
 - the most complete hadronic simulation kit on the market
 - alternative and complementary models
 - it is possible to mix-and-match, with fine granularity
 - data-driven, parameterised and theoretical models
- Consequences for the users
 - no more confined to the black box of one package
 - the user has control on the physics used in the simulation, which contributes to the validation of experiment's results

Parameterised and data-driven hadronic models

Based on experimental data

- Some models originally from GHEISHA
 - completely reengineered into OO design
 - refined physics parameterisations
- New parameterisations
 - pp, elastic differential cross section
 - nN, total cross section
 - pN, total cross section
 - np, elastic differential cross section
 - π N, total cross section
 - π N, coherent elastic scattering

p elastic scattering on Hydrogen



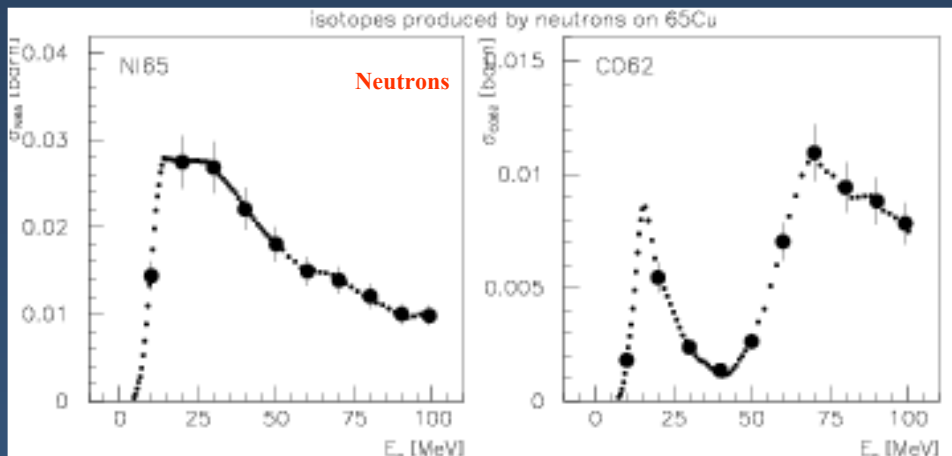
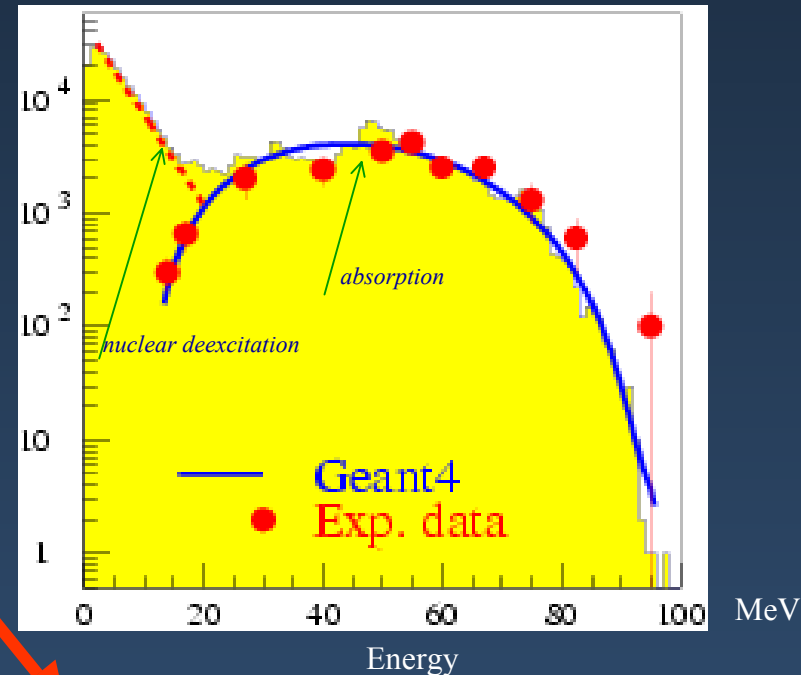
Parameterised and data-driven hadronic models

Other models are completely new, such as:

stopping particles: π^- , K^-
(relevant for μ/π PID detectors)

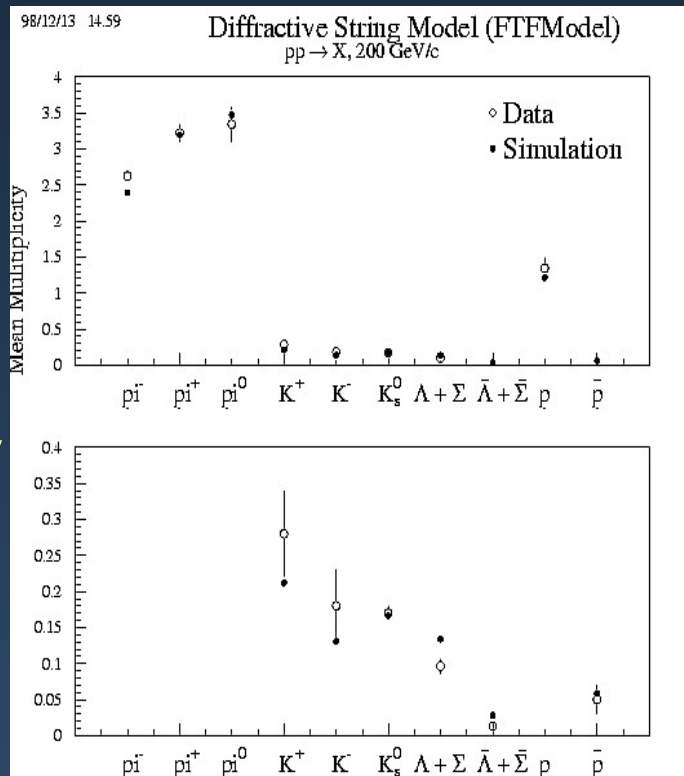
Isotope production

neutrons



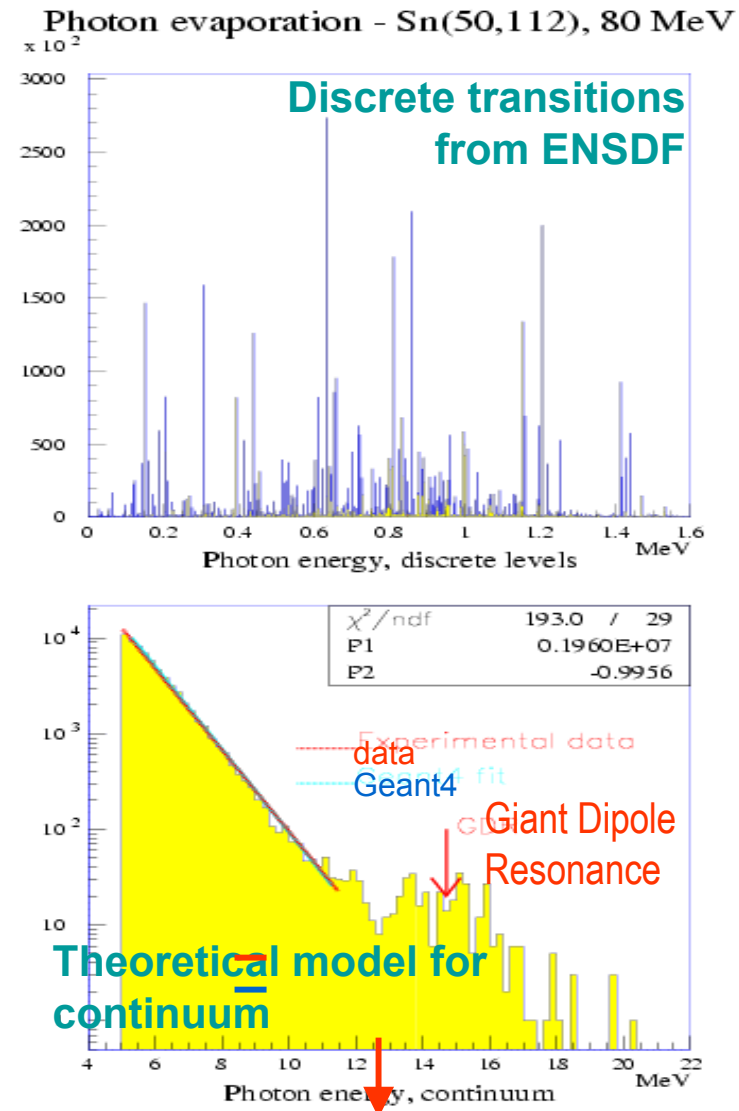
All worldwide existing databases used in neutron transport
Brond, CENDL, EFF, ENDFB, JEF, JENDL, MENDL etc.

Theory-driven models



Complementary and alternative models

- *Evaporation phase*
- Low energy range, *pre-equilibrium*, $O(100 \text{ MeV})$
- Intermediate energy range, $O(100 \text{ MeV})$ to $O(5 \text{ GeV})$, *intra-nuclear transport*
- High energy range, *hadronic generator* régime



Fast simulation

Geant4 allows to perform full and fast simulation in the same environment

- The parameterisation process produces a direct detector response, from the knowledge of particle and volume properties
 - hits, digis, reconstructed-like objects (*tracks, clusters etc.*)
- Great flexibility
 - activate fast /full simulation by detector
example: full simulation for inner detectors, fast simulation for calorimeters
 - activate fast /full simulation by geometry region
example: fast simulation in central areas and full simulation near cracks
 - activate fast /full simulation by particle type
example: in e.m. calorimeter, e/γ parameterisation + full simulation of hadrons
 - parallel geometries in fast/full simulation
example: inner and outer tracking detectors distinct in full simulation, but handled together in fast simulation

Event biasing

- Geant4 provides facilities for event biasing
- The effect consists in producing a small number of secondaries, which are artificially recognized as a huge number of particles by their statistical weights
- Event biasing can be used, for instance, for the transportation of slow neutrons or in the radioactive decay simulation
- Various variance reduction techniques available

Interface to external tools in Geant4

Through abstract interfaces

no dependence

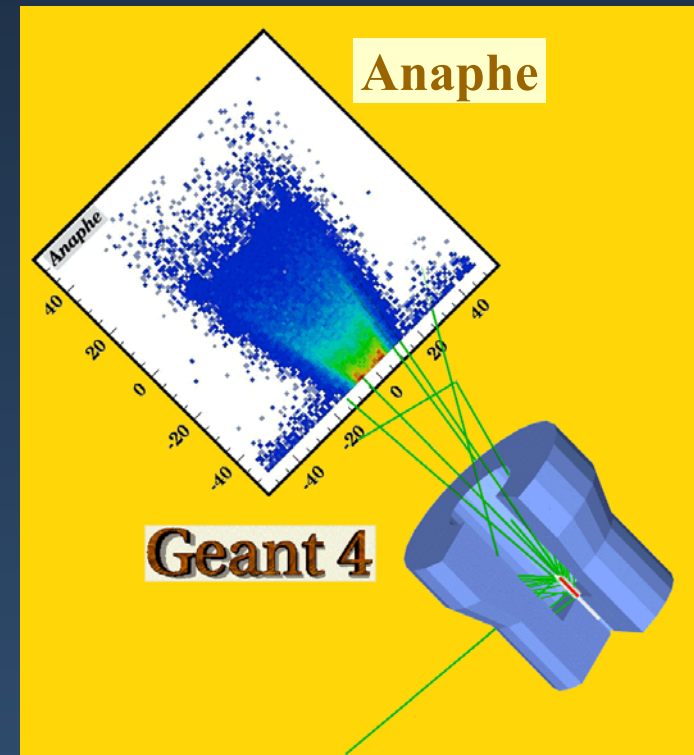
minimize coupling of components



Similar approach

- Visualisation
- (G)UI
- Persistency
- Analysis

The user is free to choose the concrete system he/she prefers for each component



More in A. Pfeiffer's talk on Analysis Tools

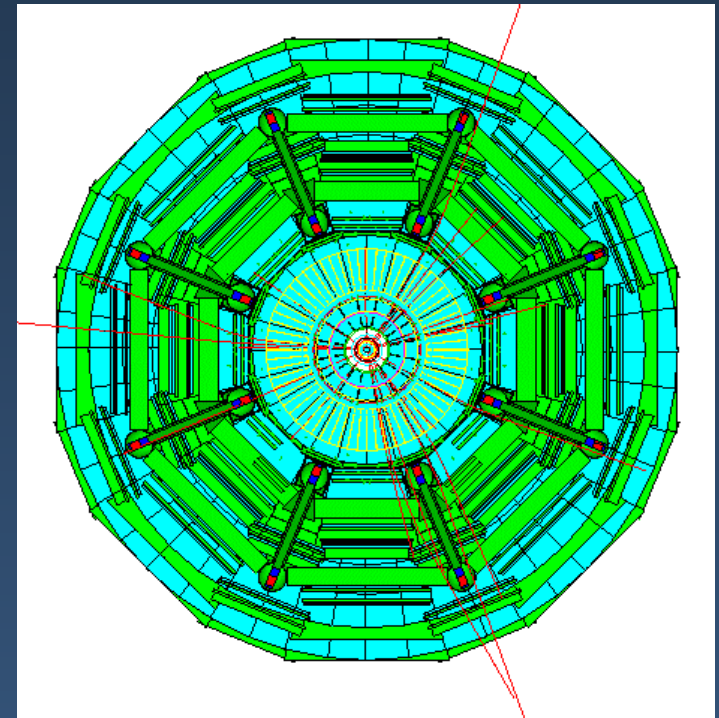
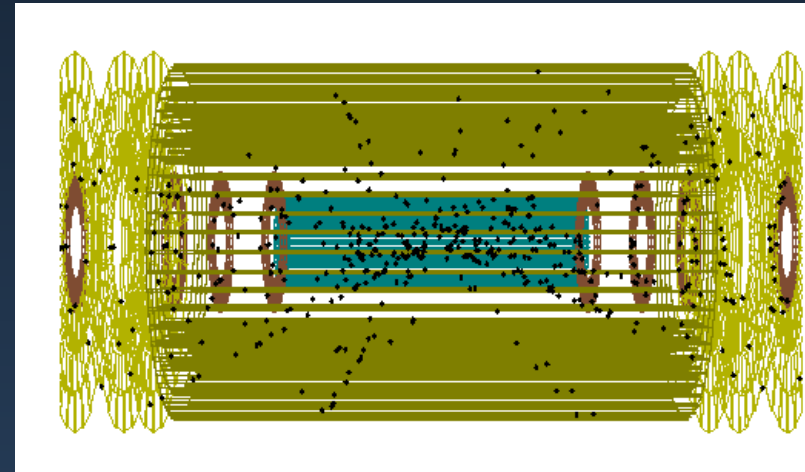
User Interface

- Two phases of user user actions
 - setup of simulation
 - control of event generation and processing
- User Interface category separated from actual command interpreter
 - several implementations, all handled through abstract interfaces
 - command-line (batch and terminal)
 - GUIs (X11/Motif, GAG, MOMO, OPACS, Java)
- Automatic code generation for geometry and physics through a GUI
 - GGE (Geant4 Geometry Editor)
 - GPE (Geant4 Physics Editor)

Visualisation

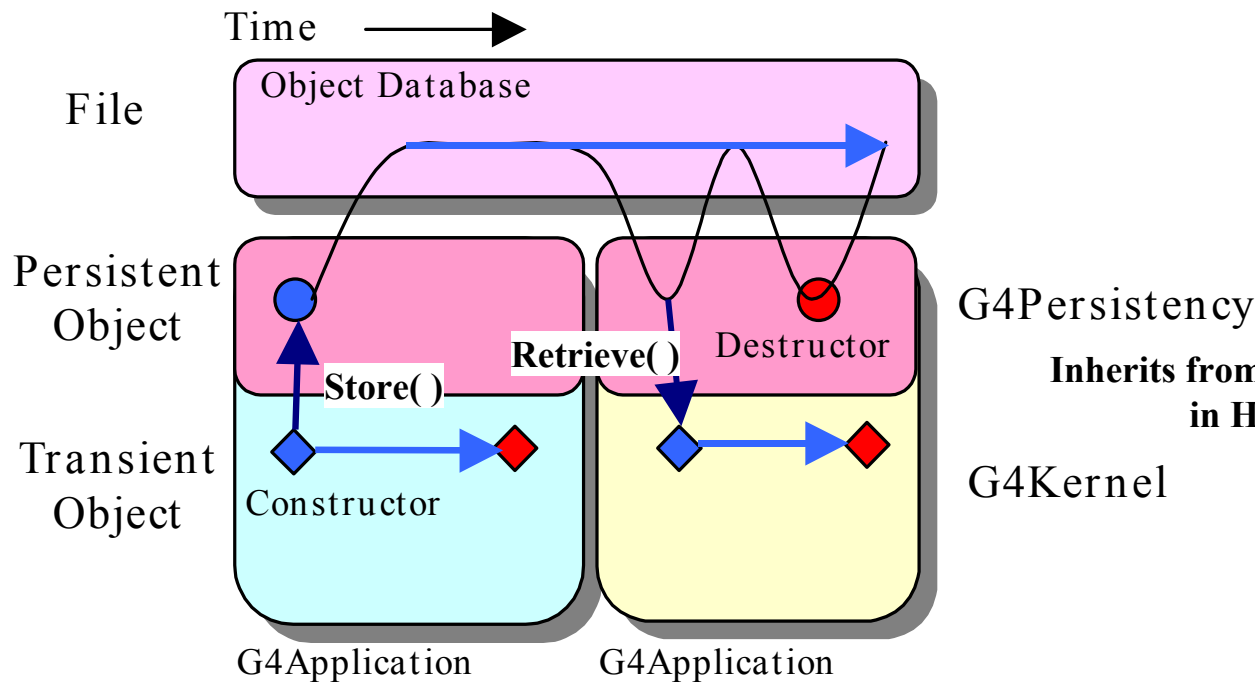
- Control of several kinds of visualisation
 - detector geometry
 - particle trajectories
 - hits in the detectors

- Various drivers
 - OpenGL
 - OpenInventor
 - X11
 - Postscript
 - DAWN
 - OPACS
 - HepRep
 - VRML...
- all handled through abstract interfaces



Persistency

- Geant4 Persistency makes run, event, hits, digits and geometry information be persistent, to be read back later by user programs
 - no dependence on any specific persistency model
 - use industrial standard ODMG C++ binding and HepODBMS as persistency interface
- Possibility to run in transient or persistent mode



G4 kernel objects have corresponding "P" objects in G4Persistency

G4Run	↔	G4PRun
G4Event	↔	G4PEvent
G4Hit	↔	G4PHit
:		:

Inherits from *HepPersObj* in HepODBMS

G4Kernel

User Documentation

- Introduction to Geant4
- Installation Guide
- User Guide - For Application Developers
 - for those wishing to use Geant4
- User Guide - For Toolkit Developers
 - for those wishing to extend Geant4 functionality
- Software Reference Manual
 - documentation of the public interface of all Geant4 classes
- Physics Reference Manual
 - extended documentation on Geant4 physics

Examples

- Novice
 - Extended
 - Advanced
- examples illustrating the main functionalities of Geant4 in realistic set-ups

Contact persons: Technical Steering Board members

(listed on the web site)